

Dialogue Processing System and Method

TECHNICAL FIELD OF THE INVENTION

The present invention relates to a dialogue processing technology for performing a dialogue with a user and for collecting any required information through the dialogue.

BACKGROUND OF THE INVENTION

A dialogue generally refers to the exchange of information. People generally use language to exchange information. Use of their voice by a person will cause the hearing sense of another person to be activated and thus information is transferred. People can also specify information on media such as paper and display device using various other means. Information is thus transferred when the sight sense of others is activated. Dialogue between people and computers or the like is very common. Computers with functions to enable such dialogue are called dialogue systems or dialogue processing systems. If the word "dialogue" is used in this patent application, it includes exchanges with computers using both the hearing and the sight senses.

In the exchange of information using language, since all required information is not necessarily included in one speech (sentence), a number of exchanges are generally required before all necessary information is acquired. In conventional dialogue systems, information is acquired using a computer led method whereby a person responds to questions asked by the computer. Therefore, the flow of the dialogue is described in advance in the system as rules. In other words, conventional dialogue systems use a flowchart model that requests the speaker to enter information using a predetermined procedure. In yet another way of looking at it, the conventional methods relate to a processing method in which the recipient of the information, that is the computer, requests the speaker to enter information.

09739644-122000

In contrast, dialogues between people cause the recipient of the information to change the behavior in accordance with the contents of the speech by the speaker. Such dialogues are therefore different to dialogues in conventional dialogue systems. Any attempts to tackle speaker led dialogue in these conventional dialogue systems necessitates the prediction of all dialogue patterns, and therefore the predication has to be performed factorial times of conditions necessary to establish a dialogue.

In Japanese Patent Application Laid-open No. 9-62739, art such as the following is disclosed. That is, in a city office, if a citizen enters their request using natural language into a device for allocating clerks at the window, the device for allocating clerks extracts keywords from the requests entered and calculates the degrees of the relationship with clerk terminals from the keywords entered. If a clerk terminal has a relationship degree higher than that of a predetermined threshold value, the device for allocating clerks connects to that clerk terminal. On the other hand, if such a clerk terminal does not exist, a vague response saying, "We cannot find a suitable clerk. Please provide more details," is returned to the citizen.

SUMMARY OF THE INVENTION

As described in the above, for conventional dialogue systems it was practically difficult to carry out a person-led dialogue, or it was impossible to carry out a natural dialogue as exchanged between persons.

Therefore, an object of the present invention is to provide dialogue processing technology that enables natural dialogue with a simple configuration.

The dialogue processing system of the present invention to perform a dialogue with a user, comprises: a plurality of slots (for example, slot 11

present invention;

Fig. 7 is a flowchart showing a specific example of knowledge base analysis and slot update processing;

Fig. 8 is a flowchart showing a specific example of knowledge base analysis and slot update processing;

Fig. 9 is a flowchart for response selection processing in the dialogue system in the preferred embodiment of the present invention;

Fig. 10 is a flowchart for slot analysis and knowledge base update processing in the dialogue system in the preferred embodiment of the present invention;

Fig. 11 is a diagram to explain a table that relates to the knowledge base in the preferred embodiment of the present invention;

Fig. 12 shows an example of a response table in the preferred embodiment of the present invention;

Fig. 13 is a diagram to explain the dialogue contents and internal status of the dialogue system in a specific example; and

Fig. 14 is a diagram to explain the dialogue contents and internal status of the dialogue system in another example.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

Firstly, the characteristics of dialogue that form the basis of the present invention, will be discussed. In a dialogue that is instituted for a specific purpose, a set of information required to establish the dialogue exists. Here, a dialogue with a specific purpose refers not to a dialogue with no specific purpose, such as a social dialogue, but to communication that is instituted for a specific reason. Also, in an airline reservation, the set of required information refers to information such as the departure point, the destination, and the flight name. There are few instances where all the information for the set of required information will be provided in one speech by the speaker. Rather, the required information is all provided through an exchange (negotiations) with the recipient. Therefore, a dialogue progresses through several repeated exchanges while situational information is omitted. Here, situational

09/23/94 11:20 AM

(1) Omission of speech contents previously discussed

(2) One piece of information can work to supplement a number of pieces of information

(3) The sequence of a dialogue cannot be predicted.

An embodiment of the present invention that was conceived on the basis of these analyses will be described below in a more specific manner.

5

connected, for example, by telephone lines. The addition to the dialogue system 100 of a microphone, speakers, and the circuit required to operate these, enables the speaker 1 to have a direct conversation with a device in which the dialogue system 100 is implemented.

The speaker 1 refers to the person that is conveying information to the dialogue system 100 in a dialogue. For example, in a dialogue system of an airline ticket reservation system, the person wishing to reserve an airline ticket is the speaker 1.

The voice processor 3 comprises a voice recognition processor 15 and a voice synthesis processor 17. The voice recognition processor 15 has a function for converting voice information entered using the voice into character strings and for outputting the character strings to the dialogue manager 5. The voice synthesis processor 17 has a function for converting character strings specified by the dialogue manager 5 into voice information and for outputting the voice information. The voice processor 3 is not required if information is input into, and output from a dialogue system 100 using character strings rather than the voice.

The dialogue manager 5 comprises a speech details analyzer 19, a dialogue processor 21, and a knowledge base processor 23. The dialogue manager 5 operates together with the script data storage unit 7, the knowledge base 9, the slot 11, and the response information storage unit 13, and has a function for indicating response information prompting the speaker 1 to enter required information and for acquiring the information required to achieve the aim of the dialogue. The speech details analyzer 19 has a function for extracting the required information from the speech information entered by the speaker 1 and for recording that extracted information into slot 11. The dialogue processor 21 has a function for acquiring response information for prompting the user to enter any information not already acquired, from the response information storage unit 13 based on information recorded in slot 11, and for outputting that acquired response information to the voice processor 3 to speak the

response information to the speaker 1. The knowledge base processor 23 has a function for analyzing the knowledge base 9 and has a function for updating the information in slot 11 and for updating the knowledge base 9 itself.

The script data storage unit 7 stores the script data that defines part of the processing in the dialogue manager 5, for example, processing to compare the information recorded in slot 11 with the information in the knowledge base 9.

The knowledge base 9 is a database that stores knowledge relating to the aim of the dialogue. For example, in an airline ticket reservation system, it stores information relating to flight schedules and reservations. More specifically, it stores information relating to the flights, such as departure points, destinations, departure times, and arrival times and information on seats available for each flight.

The items of the information required to achieve the aim of the dialogue are preset to slot 11 and slot 11 has storage areas that respectively correspond to these items of the information. By referring to these storage areas, the systems is able to grasp the state of the dialogue and is able to determine whether or not there is any other information required to achieve the aim of the dialogue. Each storage area is also called a slot.

The responses given to speaker 1 are stored in the response information storage unit 13. These responses are, specifically, sentence structure patterns, that is templates. The responses that prompt the speaker 1 to enter the information required to achieve the purpose of the dialogue are stored in accordance with the state of slot 11. The response information stored in this response information storage unit 13 can be stored as voice information or as character strings.

Next, the processing that takes place in the dialogue system 100 in the

embodiment of the present invention will be explained. Fig. 2 is a flowchart that shows the flow of whole processing that takes place in the dialogue system 100 in the embodiment of the present invention.

The speaker 1, by telephoning a telephone number allocated to the dialogue system 100, starts a dialogue with the dialogue system 100 (step S1). For example, if this dialogue system 100 is used in an airline ticket reservation system, a message saying, "Welcome. How can we help you?" will be sent to the speaker via the telephone line from the dialogue system 100. To enable this type of message to be sent from the dialogue system 100, the dialogue processor 21 selects the above response from the responses stored in the response information storage unit 13 as the message to be communicated at the beginning of a dialogue. The dialogue manager 5 communicates this selected response to the voice synthesis processor 17. The voice synthesis processor 17 converts the response sentence made from character string information into voice information. The converted voice information is then sent to the telephone of speaker 1 via telephone lines and that message is delivered to the speaker.

If, for example, in response to this the speaker 1 says on the telephone, "I would like to make a reservation on a flight to Haneda," that voice information is sent to the voice processor 3. The voice recognition processor 15 in the voice processor 3 uses well-known voice recognition processing to convert this voice information into character string information. This character string information is then entered into the dialogue manager 5 (step S3).

The speaker details analyzer 19 analyzes the contents of the speech and performs slot update processing (step S5). That is, for the character string information entered into dialogue manager 5, morpheme analysis, parsing, information extraction, and slot update processing are performed to extract the necessary information from the entered character string information. The processing involved in this step is

explained later.

After analysis of the speech contents, the knowledge base processor 23 executes knowledge base analysis and slot update processing (step S7). Here, for example, the information recorded in slot 11 is compared with the information in the knowledge base 9 based on the script data stored in the script data storage unit 7. The processing involved in this step is explained later.

Next, based on information in the updated slot 11 (step S9), the dialogue processor 21 selects a corresponding response sentence from the response stored in the response information storage unit 13. The dialogue processor 21 delivers information relating to the selected response to the voice synthesis processor 17. The information delivered to the voice synthesis processor 17 is converted into voice information and sent to the telephone of speaker 1 via the telephone line. In this way, the speaker 1 listens to the response from the dialogue system 100.

Analysis of slot 11 is then executed in the dialogue processor 21 (step S11). The analysis of slot 11 is executed from at least two viewpoints. Firstly, slot 11 is analyzed to see whether or not the information required to achieve the aim of the dialogue has been acquired. In the example of the airline ticket reservation, this analysis determines whether or not information relating to the boarding date and flight name that is required for the reservation has been acquired. If it is judged that the information required to achieve the aim of the dialogue has not all been acquired and that the dialogue is not complete (step S13), the dialogue processor 21 waits for the speaker to speak again (step S3).

If the dialogue processor 21 judges that it acquires the information required to achieve the aim of the dialogue and that the dialogue is complete (step S13), it uses the knowledge base processor 23 to update the knowledge base 9 (step S15). This is the second viewpoint from which analysis takes place, that is analysis of whether or not the knowledge

base 9 needs to be updated. In the example of the airline ticket reservation, the dialogue processor 21 judges that it acquires the information required for an airline ticket reservation from the speaker 1, and the confirmation of the reservation is acquired from the speaker 1, based on the analysis of the slot 11, the knowledge base processor 23 updates the reservation information in knowledge base 9 to reflect the information relating to the speaker's reservation to the knowledge base 9. In addition, the available seat information is updated to reduce the number of available seats by the number just reserved. This ends the processing for the dialogue with the speaker (step S17). The knowledge base 9 is updated in accordance with the aim of the dialogue. In other words, if the aim of the dialogue is a reservation, it reduces the number of available seats. If the purpose of the dialogue is to cancel a reservation, it increases the number of available seats.

Next, speech details analysis and slot update processing (step S5 in Fig. 2) will be explained in detail. Fig. 3 is a flowchart that shows the flow of processing involved in speech details analysis and slot updates. Fig. 4A and 4B show a specific example for explaining the processing involved in speech details analysis. In Fig. 4A, the arrows show the direction in which a word is subordinate.

As explained using Fig. 2, speech details analysis and slot update processing starts after voice information is received from the speaker 1 via telephone lines and is converted to character string information by the voice recognition processor 15 (step S19). This speech details analysis and slot update processing is executed in the speech details analyzer 19 of the dialogue manager 5. In this processing, morpheme analysis is the first processing to be executed (step S21). Morpheme analysis processing refers to processing that separates the input sentence into morphemes. In English, the boundary between words is clear but there is great ambiguity in some words (for example, like) that are written the same. Therefore, the analysis of the parts of speech of morphemes is main processing. In the example shown in Fig. 4A, "I want

to reserve an aircraft of Tokyo line," the sentence is divided up as, "I / want to / reserve / an aircraft / of Tokyo line." Here, "/" indicates the separation between morphemes.

Next, syntax analysis (parsing) processing is executed (step S23). Parsing processing refers to processing that generates syntax information from the results of morpheme analysis. In English, grammatical restrictions such as the positional relationship between words and parts of speech are used to analysis what relates to what. In the example in Fig. 4A, parsing shows that "I" relates to "want to", "want to" relates to "reserve", "an aircraft" relates to "reserve" and "of Tokyo line" relates to "an aircraft". As the case element, the main predicate "reserve" strongly seeks the object. Therefore, the object of "reserve" can be interpreted as that which is to be reserved. Simultaneously, "of Tokyo", which links to "an aircraft" to modify it, is used as an attribute of the object.

Information extraction processing (step S25) is executed after parsing (step S23). This information extraction processing extracts the necessary information from the results of parsing. In English, the information of the case (obligatory case) that the verb, which is the main predicate, seeks can be used. The information that indicates what case is required by a verb is already described in the dictionary. By referring to the morpheme relationships found in parsing and to obligatory case information assists in calculating the case relationship between nominal clauses that relate to the verb. For example, the fact that, "The verb 'reserve' requires an object as a case," is defined with the expression, Verb(reserve) = Case(object). Fig. 4A is, for example, converted into a case analysis graph as shown in Fig. 4B. From this, we can see that the object of "reserve" is "an aircraft" and its attribute is "of Tokyo line". If case prepositions such as to, at, in, and on are used, the information which is represented on the surface is used to extract time and place etc.

Information extraction processing (step S25) will be explained further below. Knowledge called extraction rules is used to extract information.

000227" 4496E260

This knowledge is constructed using language characteristics that appear in sentences and are processing objects. Here, as an extraction rule that extracts information from its relationship with a verb, a rule used if a call omits a case preposition is omitted in the speech is shown below. In addition, the surface information is used to extract information relating to time and etc., for example. The codes used in the following example are "Morph", which refers to a morpheme, "Next", which refers to the next morpheme, and "Object", which refers to an object word. Accordingly, Morph (reserve) = TRUE means that the morpheme is "reserve".

An example of an extraction rule using the relationship with a verb is given below.

- Table 1 -

```
if (Morph(reserve) = TRUE & Object(Morph(reserve)) = LOCATION)
    Slot(Location|arrival) ← Object(Morph(reserve))
Endif
```

This extraction rule shows that if the morpheme is "reserve" and the object for "reserve" is a word that expresses a location, information for the object of "reserve" is stored in the destination area in slot 11.

An example of an extraction rule that uses the surface information is shown below.

- Table 2 -

```
if (Morph(January) = TRUE)
    Slot(Month) ← 1
elseif (Morph(February) = TRUE)
    Slot(Month) ← 2
elseif (Morph(March) = TRUE)
    Slot(Month) ← 3
elseif (Morph(April) = TRUE)
    Slot(Month) ← 4
```

```

elseif (Morph(May) = TRUE)
    Slot(Month) ← 5
elseif (Morph(June) = TRUE)
    Slot(Month) ← 6
elseif (Morph(July) = TRUE)
    Slot(Month) ← 7
elseif (Morph(August) = TRUE)
    Slot(Month) ← 8
elseif (Morph(September) = TRUE)
    Slot(Month) ← 9
elseif (Morph(October) = TRUE)
    Slot(Month) ← 10
elseif (Morph(November) = TRUE)
    Slot(Month) ← 11
elseif (Morph(December) = TRUE)
    Slot(Month) ← 12
endif

```

This extraction rule shows that if the morpheme is a word that expresses a month between January and December, the corresponding number from 1 to 12 is stored in the month storage area in the slot 11.

An example of an extraction rule relating to destination is shown below.

- Table 3 -

```

if (Next(Morph(from)) = LOCATION)
    Slot(Location|departure) ← Next(Morph(from))
elseif (Next(Morph(to)) = LOCATION)
    Slot(Location|arrival) = Next(Morph(to))
endif

```

This extraction rule shows that if the morpheme is "from" and the word following "from" is a location, then the word following "from" will be stored in the departure point storage area of the slot 11. Also, if the

morpheme is "to" and the word following "to" is a location, then the word following "to" will be stored in the destination storage area of the slot 11.

Information is thus extracted (step S25) and slot 11 is updated (step S27).

The flow of processing involved in information extraction (step S25) will be further explained using the flowchart shown in Fig. 5. After parsing processing, the speech details analyzer 19 starts information extraction (step S31). Lastly, the speech details analyzer 19 applies the above types of extraction rules and executes processing (step S33). The analyzer determines whether or not the extracted information relates to the departure point (step S35). If it does relate to the departure point, the speech details analyzer 19 sets that information in the departure point storage area in slot 11 (step S37). If the information does not relate to the departure point, the analyzer 19 determines whether or not it relates to the destination (step S39). If it does relate to the destination, the analyzer 19 sets the information in the destination storage area in slot 11 (step S41).

If the extracted information does not relate to the destination, the analyzer 19 determines whether or not it relates to the date (step S43). If the information does relate to the date, the analyzer 19 sets it into the date storage area in slot 11 (step S45). If the information does not relate to the date, the analyzer 19 determines whether or not it relates to the time (step S47). If it does relate to the time, the analyzer 19 sets it in the time storage area (step S49). If the extracted information does not relate to time, the analyzer 19 determines whether or not it relates to the flight name (step S51). If it does relate to the flight name, the analyzer 19 sets it in the flight name storage area in slot 11 (step S53). If it does not relate to the flight name, the analyzer 19 determines whether or not the extracted information indicates the end of the syntax (parsing) information that should be analyzed (step S55). If it is not the end of the syntax information, the analyzer 19 applies the extraction rules again to repeat the above processing (step S33). If it is the end of the syntax

information, information extraction ends (step S57).

Information extraction and slot update processing is thus executed and processing ends (Fig. 3: step S29).

Next, details of knowledge base analysis and slot update (step S7 in Fig. 2) will be explained. Fig. 6 is a flowchart that shows the flow of processing involved in knowledge base analysis and slot update processing.

In Fig. 6, processing of knowledge base analysis and slot update starts by the knowledge base processor 23 after processing of speech details analysis and slot update (step S67). This knowledge base analysis and slot update processing includes processing to verify the conformity between information recorded in slot 11 and the actual situation and processing to supplement necessary information in accordance with the information recorded in slot 11 and the information stored in the knowledge base 9. Here, in the example of the flight reservation, the actual situation refers to the flight schedules and reservation status. Firstly, script data is read from the script data storage unit 7 to compare the information in the knowledge base 9 with that in the slot 11 (step S69).

Next, information stored in the predetermined area of the slot 11 through speech details analysis and slot update processing (step S5 in Fig. 2) is read out (step S71). The knowledge base processor 23 uses the script data read from the script data storage unit 7 to compare the information read from the slot 11 with the information in the knowledge base 9 (step S73). The slot will be updated as necessary depending on the results of the comparison (step S74). If the processing relating to the script data has not ended (step S75), script data will be read again from the script data storage unit 7 (step S69) and processing will continue as above. If the script data has ended (step S75), knowledge base analysis and slot update processing ends (step S77).

As an example of processing that is included in the knowledge base analysis and slot update processing, the processing that verifies whether or not any contradiction exist between the information set in the slot 11 and the information in the knowledge base 9 will be explained. The script for this verification will be explained using the example of the airline ticket reservation. To execute this script, the knowledge base 9 has a function for acquiring the "departure point" and "flight name" information stored in the slot 11 and for outputting TRUE if verification is successful and FALSE if there is a contradiction. An example of the script is given below.

- Table 4 -

```
if(Knowledge(Slot(departure point)|Slot(flight name)) = FALSE)
    Slot(departure point) ← Not relevant
endif
```

In this script, the "departure point" and "flight name" stored in the slot 11 are given to the knowledge base 9 and its return value is verified. If there is any contradiction in the "departure point" and "flight name", a code that indicates a contradiction, for example "not relevant" will be set in the departure point storage area in the slot 11 to update the slot 11. When the dialogue processor 21 selects a response from the response information storage unit 13, updating the slot 11 in this way causes a code indicating a contradiction in the slot 11 and enables the selection of a response that indicates that there is an inconsistency or an error in the speech contents by the speaker 1.

Next, processing included in the knowledge base analysis and slot update processing will be explained, that is for deriving further necessary information based on the information extracted from the speech details analyzer 19 and for storing the derived information in the predetermined storage area in the slot 11.

The script that derives the related information and stores it in the slot 11

will be explained, once again using the example of the airline ticket reservation. Reservations for an airplane can be made using departure point, destination, date, and time information but the flight name is information for which departure points, destination, and time attributes are uniquely set. In other words, by specifying the flight name, departure points, destinations, and times are also specified. Using this characteristic, if flight name information is extracted in the knowledge base analysis, processing to automatically supplement the destination slot, the departure point slot and time slot is executed. This type of knowledge base analysis and slot update processing can be used to reduce the number of questions that need be asked for the speaker 1.

It is also possible to use the departure point, destination, and time information to supplement the flight name information. Furthermore, if there is only one flight a day from a certain airport A to a certain airport B, information relating to the departure point and destination can be used to supplement the flight name. The script that executes such supplementary processing is called supplementary script.

Next, an example of this supplementary script will be explained. Firstly, the supplementary script that uses the knowledge base 9 to supplement information relating to departure point, destination and time if the date and flight name information have been able to be extracted, will be explained. The code "Knowledge" used in this supplementary script refers to the knowledge in the knowledge base 9.

- Table 5 -

```
if (Slot (date) = TRUE & Slot (flight name) = TRUE)
```

```
    Slot (departure point) ← Knowledge (departure point | flight
    name)
```

```
    Slot (destination) ← Knowledge (destination | flight name)
```

```
    Slot (time) ← Knowledge (time | flight name)
```

```
endif
```

If the date and flight name information are included in the slot 11, this

supplementary script stores the departure information in the departure point storage area in the slot 11 (departure point slot) using the knowledge that links the flight name and departure points in knowledge base 9. In addition, it stores the destination information in the destination storage area of the slot 11 (destination area slot) using the knowledge that links the flight name and destination in the knowledge base 9. Furthermore, using the knowledge that links the flight name and time in knowledge base 9, the script stores the time information in the time storage area of the slot 11 (time slot). The date information is sometimes omitted because it is not essential for acquiring the departure point, destination, and time information. Such processing eliminates the need to ask the speaker questions relating to the departure point, destination, and time and thus enables more efficient and speedy acquisition of the information needed for an airline ticket reservation, which is the aim of the dialogue.

Next, an example of the supplementary script that executes the processing for providing flight names from the knowledge base 9 if the departure point, destination, and time information have been able to be extracted, will be explained.

- Table 6 -

```
if (Slot (date) = TRUE & Slot (departure point) = TRUE & Slot
(destination) = TRUE & Slot (time) = TRUE)
```

```
    Slot (flight) ← Knowledge (flight name | date, departure point,
destination, time)
endif
```

If the slot 11 includes date, departure point, destination, and time information, this supplementary script describes the processing for storing a flight name information in the flight name storage area of the slot 11 (flight slot) based on knowledge from the knowledge base 9 that links date, departure point, destination, and time information. Such processing eliminates the need to ask questions relating to the flight and

thus enables more efficient and speedy acquisition of the information needed for an airline ticket reservation.

The next example is one of supplementary script that executes the processing for providing the flight name from information relating to a departure point and destination if there is only one flight per day from a certain airport A to a certain airport B.

- Table 7 -

```
if (Slot (date) = TRUE & Slot (departure point) = TRUE & Slot
(destination) = TRUE & Knowledge (candidates | date, departure point,
destination) = 1)
```

```
    Slot (flight) ← Knowledge (flight name | date, departure point,
destination)
```

```
endif
```

If the slot 11 includes date, departure point, and destination information and if there is only one flight for the date, departure point, and destination provided, this supplementary script describes the processing for storing flight name information in the flight name storage area of the slot 11 based on the knowledge in the knowledge base 9 that links the date, departure points, and destinations. Such processing eliminates the need to ask the speaker 1 questions relating to the flight name and thus enables more efficient and speedy acquisition of the information needed for an airline ticket reservation, which is the aim of the dialogue.

Next, a specific flow of processing involved in knowledge base analysis and slot update if a date and flight name have been specified will be explained using Fig. 7. First, the knowledge base processor 23 reads the script that supplement the departure points, destinations, and times if date and flight name information have been extracted, from the script data storage unit 7 (step S79). The processing enclosed by the dotted line in Fig. 7 is executed in accordance with the script description. Next, the knowledge base processor 23 reads information in the slot 11, that is

the information extracted from the speech information that has been input by the speaker 1 (step S81). Note that the execution order of the steps S79 and S81 can be reversed. Later processing is then executed in accordance with the script read in step S79.

Whether or not date information is included in the information read from the slot 11 is determined by the knowledge base processor 23 in accordance with the script (step S83). Processing ends if date information is not included (step S93). If date information is included, whether or not flight name information is included in the read information is determined (step S85). Processing ends if flight name information is not included (step S93).

If flight name information is included, tables that link departure point, destination, and time information to the date and flight name information, stored in the knowledge base 9 is referred to, and the departure point, destination, and time information corresponding to this date and flight name information is read out. The read departure point information will be placed in the departure point slot (step S87). The destination information will be placed in the destination slot (step S89), and the time information read will be placed in the time slot (step S91). If this information has been placed in the slot 11, the sequence of knowledge base analysis and slot update processing will end. In this processing flow, which has been explained using Fig. 7, inclusion of date information was the prerequisite of knowledge base processing. However, if flight name information is included and departure point, destination, and time information can be acquired from the knowledge base 9 by the flight name information, date information is not essential.

Next, Fig. 8 will be used to explain the specific flow of processing involved in knowledge base analysis and slot update if there is only one flight per day from a certain airport A to a certain airport B and if the flight name is supplemented from information relating to the departure point and destination. Firstly, the script stored in the script data storage unit 7 to

people for which reservations can be made is given to this slot Slot (status), information saying, "There are 100 seats available." can be provided as a response. A template for a response would be configured as follows.

- Table 9 -

There are "Slot (status)" seats available.

" " is a symbol that indicates the information stored in the slot.

Next, selection of the response in Fig. 2 (step S9) will be explained. Fig. 9 is a flowchart showing the flow of processing that relates to this response selection. Firstly, the information stored in the slot 11 is read (step S107). The response table from the response information storage unit 13 is also read (step S109). Note that the execution order of the steps S107 and S109 can be reversed. A plurality of combinations of slot statuses and responses are recorded in this response table. The dialogue processor 21 selects a response based on the information read from the slot 11, that is, selects a response from the response table in accordance with the state of slots being filled (step S111). The dialogue processor 21 then outputs the character string information for the selected response to the voice synthesis processor 17 (step S113). This character string information is converted into voice information in the voice synthesis processor 17 and transmitted via a telephone line to the speaker 1. Then, this sequence of processing for response selection ends (step S115).

Next, the details of the slot analysis and knowledge base update processing in Fig. 2 (steps S11 through S15 in Fig. 2) will be explained using Fig. 10. Firstly, the knowledge base processor 23 reads information stored in the slot 11 (step S117). The knowledge base processor 23 then determines whether or not the dialogue has finished based on the read information (step S119). If it determines that the dialogue has not finished, it waits for speech input (step S121). In other words, processing moves to the speech receiving step (step S3) in Fig. 2.

If it determines that the dialogue has finished, it determines whether or not the knowledge base needs to be updated (step S123). Processing ends if it judges that the knowledge base need not be updated (step S135).

If the knowledge base processor 23 judges that the knowledge base 9 needs be updated, it executes processing to update the knowledge base 9 (step S125). In the knowledge base update processing in this example, the knowledge base processor 23 first determines that the number of available seats must be reduced (step S127). In other words, an airline ticket reservation has been executed by information stored in the slot 11 and then, the knowledge base processor 23 judges that the number of available seats must be reduced by the number of tickets just reserved. If the number of available seats must be reduced, processing is executed to reduce the number of available seats for the flight for which reservations have been received (step S129). The result of the processing to reduce the number of available seats is reflected in the information relating to available seats in the knowledge base 9.

If there is no need to reduce the number of available seats, the knowledge base processor 23 determines whether or not there is a need, conversely, to increase the number of available seats (step S131). The number of available seats may need to be increased if, for example, a reservation is cancelled. If the knowledge base processor 23 determines that the number of available seats needs to be increased, it executes processing to increase the number of available seats (step S133). The processing to increase the number of available seats is executed for the knowledge base 9. Processing ends if the knowledge base processor 23 judges that there is no need to increase the number of available seats (step S135).

The specific configuration of the knowledge base 9 will be explained next. Fig. 11 is an example of a knowledge base relating to airline ticket reservations. Fig. 11 shows information about flights departing from Haneda airport between 1000 hours and 1850 hours on June 5. This information includes date, departure time, departure point, destination,

flight name, and available seat information. In its actual operation, in addition to information relating to Haneda as a departure point, information of flights departing from various airports is included. In this knowledge base 9, the date, time, departure point, destination, and flight name information are linked and so, for example, if a flight name is specified, the departure times, departure points, and destination information can be retrieved. Also, because the number of available seats is given for each flight, if the dialogue system 100 in the embodiment of the present invention is used to reserve an air ticket or to cancel a reservation, the number of available seats can be either reduced or increased.

Next, an example of a response table will be explained using Fig. 12. Fig. 12 is divided into a column relating to the slot status and a column for responses. Information such as date, flight, departure point, and destination can be set in the slot status column. The parts in this figure marked with "O" refer to a status in which information has been set in the slot. The parts marked with "X" refer to a status in which information has not been set in the slot.

If no information has been set, the system is configured to output a response saying, "Please specify the flight you would like to board." The system is also configured so that a response stating, "Please specify your destination," will be output if only the date information is set, and a response stating, "Please specify your departure point and destination," will be output when a date and flight have been set. Note, however, that if a script such as that described above is used, and if the date and flight name information is set, a response statement need not be prepared. The system is further configured so that a response stating, "Please specify your departure point," will be output if the date and destination are specified. If all four items of information are set, a response stating, "Thank you for your reservation," will be output. Responses are also specified according to slot status. As described above, responses contain a question that will lead to a status whereby the necessary information

processor 21 refers to the information in this slot 11, selects a response from the response information storage unit 13, and outputs this to the voice synthesis processor 17. In this case, a response stating, "Please specify your departure point," is selected.

If the speaker 1 enters, "I would like a flight leaving Tokyo in the morning," the speech details analyzer 19 in the dialogue system 100 extracts that the departure point is "Tokyo" and stores this in the storage area in the slot 11 relating to the departure point (departure point slot). It also extracts the information "in the morning" and the knowledge base processor 23 analyzes the knowledge base 9 with "in the morning". Based on information stored in the slot 11 so far, the system recognizes that the flight is to depart Tokyo for Fukuoka during the morning of June 6. Then, information from the knowledge base 9 representing that there is only one flight fulfilling these requirements and departing at 1100 hours is acquired. Based on the acquired information, information representing that the flight is the one departing at 1100 hours is stored in the flight name storage area (flight slot) of the slot 11. The dialogue processor 21 selects the response stating, "There is an 11 am flight," in accordance with the information stored in the slot 11, and outputs this to the speaker 1. If the speaker 1 enters, "That will be fine," in response, the dialogue system 100 outputs a response, "Thank you very much," and ends the sequence of dialogue processing. It is also possible to provide another slot that stores the fact that a confirmation of reservation has been received, and if information is stored in this slot, a response stating, "Thank you for very much" may be output. Furthermore, the system can be configured so that after confirmation that information has been recorded into the slot for recording the fact that a reservation has been confirmed, update processing for the knowledge base 9, that is processing to reduce the number of available seats, is executed.

In the example shown in Fig. 13, "in the morning" information was entered and now the processing for this will be further explained. The information "in the morning" differs from date information in that it has

000022T"4496E/60

a time-space range. Therefore, a response using just the extraction rules is not possible and a check must be made to the knowledge base 9. Here, an example is shown where extraction rules and a script for analyzing the knowledge base and updating the slot are prepared. To simplify explanations, the flight slot shown in Fig. 13 is not used and another range slot, which is the storage area for storing the information relating to the time range known as "in the morning," is prepared. In the actual implementation of a dialogue system 100, it is necessary to limit the number of slots as much as possible. This is to prevent an explosion in the number of possible response patterns. Accordingly, the flight slot could also be shared with the range slot as shown in Fig. 13. In this kind of situation, the following extraction rule could be used.

- Table 10 -

```
if (Morph ("in the morning") = TRUE)
    Slot (range) ← "in the morning"
endif
```

This extraction rule specifies processing for storing information "in the morning" in the range slot of the slot 11 if the morpheme is "in the morning".

Furthermore, in knowledge base analysis and slot update processing, processing on these contents of slot 11 is executed with the following script.

- Table 11 -

```
if (Slot (range) = "in the morning")
    Slot (candidate) ← Knowledge (list | Slot (date), Slot (departure
point), Slot (destination), first departure to 1200 hours)
endif
```

This script shows that if information "in the morning" is stored in the range slot of slot 11, the knowledge in knowledge base 9 is used in the

processing to store the list of information fulfilling the requirements relating to the date, departure point, destination, and departure times from the first departure to 1200 hours found in slot 11 in the storage area for candidates in slot 11 (candidate slot). Thus, if the time range is defined in Slot (range), a time range, as well as date, departure point, and destination elements can be defined and referred to. This enables a list of relevant flights to be given in the Slot (candidates).

By setting a response asking, "There are "Slot (candidate)" flights available. Which would you prefer?" for this list, the speaker 1 can be made to select the flight they prefer from the list of possible flights. The area enclosed by "" refers to information itself stored in the slot.

Now Fig. 14 will be used to further explain the sequence of the contents of a dialogue and the internal status of the dialogue system 100 that relates to the embodiment of the present invention. The speaker column does not contain simply the source of the call but also the processing entity. If the speaker 1 first makes a call on the telephone to say, "I would like to make a flight reservation," that information is delivered to the dialogue system 100. The dialogue processor 21 selects from the response information storage unit 13 the response "Please specify the flight you would like to board," based on the results of analysis by the speech details analyzer 19, and outputs the response to speaker 1 via the voice synthesis processor 17.

When the speaker 1 says, "I would like to go to Fukuoka," the speech detail analyzer 19 extracts information that the destination is "Fukuoka", and the information "Fukuoka" is then stored in the destination slot of slot 11. The dialogue processor 21 refers to the information in the slot 11, selects a response from the response information storage unit 13, and outputs it. In this case, a response "When would you like to board the flight?" will be selected.

When the speaker 1 enters "Tomorrow morning, please" in response to

000222T 4496Z69

this, the speech details analyzer 19 in the dialogue system 100 extracts the date "tomorrow" and stores this in the storage area for the date in slot 11. To complete the reservation, this information "tomorrow" must be converted to an absolute value. To do this, the date information of the dialogue is acquired from the timer in the dialogue system 100 and that value of the date information is changed into that value plus one day. This is stored in the date slot of slot 11. In this example, the date of the dialogue is the 7-th and so the 8-th, which is 7+1, is stored in the date slot.

Here, the speech details analysis if information "tomorrow" has been entered will be explained. The processing here is executed in accordance with the following extraction rule.

- Table 12 -

```
if (Morph ("tomorrow") = TRUE)
    Slot (date) ← Date ("today" + 1)
endif
```

This shows the processing wherein date information, in which 1 is added to the current date, is store in the date slot of slot 11 if tomorrow is the morpheme.

The extraction rule for "the day after tomorrow" is shown below.

- Table 13 -

```
if (Morph ("the day after tomorrow") = TRUE)
    Slot (date) ← Date ("today" + 2)
endif
```

Also, for information extracted as "in the morning", the knowledge base processor 23 analyzes the knowledge base and updates the slots. In order to show the flights from the dialogue system 100 to the speaker 1, the system 100, then, recognizes that a departure point must be specified.

Therefore, the dialogue processor 11 selects the response "Please specify your departure point" from the response information storage unit 13 and outputs this to the speaker 1. If the speaker 1 enters "I would like a flight departing from Tokyo", information is extracted in the speech details analyzer 19 and the information "Tokyo" is stored in the departure point slot in slot 11.

Based on the information stored in slot 11 acquired thus far, it is recognized that the flight that the speaker 1 is attempting to reserve is the flight departing in the morning on June 8 from Tokyo and going to Fukuoka. Therefore, the knowledge base processor 23 searches the knowledge base 9 and acquires information that the only flight fulfilling all requirements is one departing at 1100 hours. The acquired information is stored in the flight name slot in slot 11. Then, in accordance with the information stored in slot 11, the dialogue processor 21 selects a response stating, "There is a flight departing at 11 o'clock," and outputs this to the speaker 1. If the speaker 1 responds with, "That will be fine," the dialogue system 100 outputs a response "Thank you very much" and ends the sequence of dialogue processing.

In the dialogue system 100 in the embodiment of the present invention, the necessary information is recorded in slots and response information is selected based on the status of those slots. Therefore, even if some of the necessary information recorded in a slot changes, there is no need to re-enter all the necessary information. For example, from the initial speech in the flight reservation system, it is easy to change from "Narita Airport," which is the first input, to "Haneda Airport".

In the above example, the speaker 1 entered information by voice into the dialogue system 100 but this is not the only means of input possible. The speaker 1 can also enter information by typing character string information using a keyboard connected to the dialogue system 100 or can send character string information via e-mail.

The function block diagram shown in Fig. 1 is one example and it is also possible to integrate a plurality of blocks into one and to divide one block into a plurality of blocks.

Furthermore, the dialogue system 100 described above can also be configured by installing a program for a dialogue system onto a computer. Here, the program would be stored on a storage medium or storage device such as a floppy disk, CD-ROM, a magneto-optical disk, semi-conductor memory, or hard disk. The results of interim processing would temporarily be stored in memory.

The present invention has thus enabled a dialogue processing system with a simple configuration to realize natural dialogue.

Although the present invention has been described with respect to a specific preferred embodiment thereof, various change and modifications may be suggested to one skilled in the art, and it is intended that the present invention encompass such changes and modifications as fall within the scope of the appended claims.

000221" 4496E/60